



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
[www.uspto.gov](http://www.uspto.gov)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/729,015	12/04/2000	Minoru Takimoto	FUJR 18.034	6757

7590                    09/25/2003

KATTEN MUCHIN ZAVIS ROSENMAN  
575 MADISON AVE  
NEW YORK, NY 10022-2585

EXAMINER

ROCHE, TRENTON J

ART UNIT

PAPER NUMBER

2124

DATE MAILED: 09/25/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

<b>Office Action Summary</b>	<b>Application No.</b>	<b>Applicant(s)</b>
	09/729,015	TAKIMOTO, MINORU
	<b>Examiner</b>	<b>Art Unit</b>
	Trent J Roche	2124

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

#### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

#### Status

1) Responsive to communication(s) filed on 04 December 2000.

2a) This action is FINAL.                    2b) This action is non-final.

3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

#### Disposition of Claims

4) Claim(s) 1-31 is/are pending in the application.

4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.

5) Claim(s) \_\_\_\_\_ is/are allowed.

6) Claim(s) 1-31 is/are rejected.

7) Claim(s) \_\_\_\_\_ is/are objected to.

8) Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

#### Application Papers

9) The specification is objected to by the Examiner.

10) The drawing(s) filed on 04 December 2000 is/are: a) accepted or b) objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11) The proposed drawing correction filed on \_\_\_\_\_ is: a) approved b) disapproved by the Examiner.  
If approved, corrected drawings are required in reply to this Office action.

12) The oath or declaration is objected to by the Examiner.

#### Priority under 35 U.S.C. §§ 119 and 120

13) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a) All b) Some \* c) None of:

1. Certified copies of the priority documents have been received.

2. Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.

3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

14) Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).  
a) The translation of the foreign language provisional application has been received.

15) Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

#### Attachment(s)

1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)	4) <input type="checkbox"/> Interview Summary (PTO-413) Paper No(s). _____ .
2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)	5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152)
3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO-1449) Paper No(s) 2 .	6) <input type="checkbox"/> Other: _____ .

## **DETAILED ACTION**

1. Claims 1-31 have been examined.

### *Priority*

2. Acknowledgment is made of applicant's claim for foreign priority under 35 U.S.C. 119(a)-(d).

The certified copy has been filed in parent Application No. 09/729,015, filed on December 4, 2000.

### *Specification*

3. Applicant is reminded of the proper language and format for an abstract of the disclosure.

The abstract should be in narrative form and generally limited to a single paragraph on a separate sheet within the range of 50 to 150 words. It is important that the abstract not exceed 150 words in length since the space provided for the abstract on the computer tape used by the printer is limited. The form and legal phraseology often used in patent claims, such as "means" and "said," should be avoided. The abstract should describe the disclosure sufficiently to assist readers in deciding whether there is a need for consulting the full patent text for details.

The language should be clear and concise and should not repeat information given in the title. It should avoid using phrases which can be implied, such as, "The disclosure concerns," "The disclosure defined by this invention," "The disclosure describes," etc.

The abstract exceeds 150 words.

### *Claim Rejections - 35 USC § 101*

4. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims 29-31 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

Art Unit: 2124

The invention as disclosed in claims 29-31 are directed to non-statutory subject matter. While the claims are in the technological arts, they are not limited to "a practical application of an abstract idea which produced a useful, concrete, and tangible result." State Street Bank & Trust v. Signature Financial Group, Inc., 149 F. 3d 1368, 1375 n. 9 (Fed. Cir. 1998).

The product recited in claims 29-31 is not explicitly shown to perform any of the claimed steps of the invention. Rather, it states that the product outputs control information which directs the server apparatus *to comprise* the steps as recited in the claims, and therefore fails to show that its function has a practical application which would produce useful, concrete, and tangible results under the State Street Formulation.

On this basis, claims 29-31 are rejected under 35 U.S.C. § 101.

To overcome this rejection, it is recommended that the claims be adjusted to distinctly point out that the product directs the program execution apparatus *to perform* the steps as recited in the claims.

### *Claim Rejections - 35 USC § 102*

5. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(a) the invention was known or used by others in this country, or patented or described in a printed publication in this or a foreign country, before the invention thereof by the applicant for a patent.

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

Art Unit: 2124

6. Claims 1, 8-10, 21, 28, 29 and 31 are rejected under 35 U.S.C. 102(a) as being anticipated by U.S. Patent 5,974,454 to Apfel et al.

**Regarding claim 1:**

Apfel et al teach:

- an information processing system (“a system for updating a software program...” in col. 14 line 8)
- at least one terminal apparatus and at least one program execution apparatus (“the computer, a database server, and a package server...” in col. 14 line 10)
- each of said at least one terminal apparatus comprises a message transmitting unit which transmits a message containing version information indicating a program version (“sending an upgrade package message from the database server to the computer...” in col. 12 lines 29-30)
- a message receiving unit which receives a message containing version information indicating a program version, from one of said at least one terminal apparatus (“sending an upgrade package message from the database server to the computer...” in col. 12 lines 29-30. There must be a receiving unit for the computer to receive the upgrade package message.)
- a program storing unit which stores one or more program components (“the personal computer further includes a hard disk drive...” in col. 4 lines 22-23, and further, “A number of program modules may be stored in the drives...” in col. 4 line 40)
- a pre-transfer information management table which holds information on said one or more program components stored in said program storing unit (“a database query is sent from the computer...the database query...can include information such as the version of the program

module component...the platform that the program module component is running on, and the languages of the program module component." in col. 2 lines 28-35. This information contained in the query must have been stored in an information table on the program storing unit.)

- a program memory unit which is allocated to an activated process, and temporarily stores at least one program component transferred from said program storing unit ("program modules may be stored in the...RAM..." in col. 4 lines 40-41)
- a post-transfer information management table which holds information on said at least one program component stored in said program memory unit ("the upgrade package is installed on computer..." in col. 11 line 1. Further, since the newly installed program must have version information, this information must overwrite the prior information that was stored in the information table for the old version of the program.
- a program executing unit which dynamically links one of said one or more program components corresponding to said version information contained in said message received by said message receiving unit, to said program memory unit, so as to enable execution of said one of said one or more program components in said process ("the upgrade package is installed on computer..." in col. 11 line 1. The installation program must link the program to a memory unit for the program to be able to operate.)

**Regarding claim 8:**

The rejection of claim 1 is incorporated, and further, Apfel et al disclose a management apparatus which sets and manages the one or more program components ("Auto-autoupdate is a cooperative

Art Unit: 2124

scheduled client 'pull' routine designed to automatically update the...program module..." in col. 3 lines 39-41. The updating client is a management apparatus for the updating of the components.)

**Regarding claim 9:**

Apfel et al teach:

- a terminal apparatus ("database server..." in col. 6 line 40)
- an interface unit which controls operations of inputting and outputting software (An interface unit is inherently present in a computer system.)
- a message transmitting unit which transmits a message containing version information indicating a program version ("sending an upgrade package message from the database server to the computer..." in col. 12 lines 29-30)

**Regarding claim 10:**

Apfel et al teach:

- a program execution apparatus ("The personal computer..." in col. 4 line 60)
- a message receiving unit which receives a message containing version information indicating a program version ("sending an upgrade package message from the database server to the computer..." in col. 12 lines 29-30. There must be a receiving unit for the computer to receive the upgrade package message.)
- a program storing unit which stores one or more program components ("the personal computer further includes a hard disk drive..." in col. 4 lines 22-23, and further, "A number of program modules may be stored in the drives..." in col. 4 line 40)

Art Unit: 2124

- a pre-transfer information management table which holds information on said one or more program components stored in said program storing unit (“a database query is sent from the computer...the database query...can include information such as the version of the program module component...the platform that the program module component is running on, and the languages of the program module component.” in col. 2 lines 28-35. This information contained in the query must have been stored in an information table on the program storing unit.)
- a program memory unit which is allocated to an activated process, and temporarily stores at least one program component transferred from said program storing unit (“program modules may be stored in the...RAM...” in col. 4 lines 40-41)
- a post-transfer information management table which holds information on said at least one program component stored in said program memory unit (“the upgrade package is installed on computer...” in col. 11 line 1. Further, since the newly installed program must have version information, this information must overwrite the prior information that was stored in the information table for the old version of the program.
- a program executing unit which dynamically links one of said one or more program components corresponding to said version information contained in said message received by said message receiving unit, to said program memory unit, so as to enable execution of said one of said one or more program components in said process (“the upgrade package is installed on computer...” in col. 11 line 1. The installation program must link the program to a memory unit for the program to be able to operate.)

**Regarding claim 21:**

Art Unit: 2124

Apfel et al teach:

- a method for updating a program component (“method for updating software program...” in col. 2 lines 13-14)
- loaded in a server in an N-tier client-server environment (“the computer, a database server, and a package server...” in col. 14 line 10)
- storing in said server one or more program components (“new program module components may also be obtained from the package server...” in col. 5 lines 25-26)
- holding information on said one or more program components stored in said server, in a pre-transfer information management table (“the database server needs to determine if an upgrade is available...” in col. 6 lines 50-51. For the server to determine if an upgrade is available, it must have a table of current versions presently stored on the server.)
- transmitting a first message containing version information indicating a program version from a client to a server (“a database query is sent from the computer over the Internet to a database server...the database query...can include information such as the version of the program module component...” in col. 2 lines 28-33)
- receiving said first message by said server (“In response to receiving the database query, the database server determines...” in col. 2 lines 36-37)
- dynamically linking one of said one or more program components corresponding to said version information contained in said first message to a memory which is allocated to an activated process in said server, so as to enable execution of said one or said one or more program components in said process (“in a networked environment, program modules depicted relative to the personal computer, or portions thereof, may be stored in the remote

Art Unit: 2124

memory storage device.” in col. 5 lines 13-16. To use the remote program module, the component would have to be linked to memory.)

- holding in a post-transfer information management table information on at least one program component stored in said memory (“the servers may also provide a ‘next check’ date if there is not an upgrade currently available.” in col. 7 lines 25-27. The date information must be stored in a table somewhere on the system for each component in memory.)
- transmitting to another server a second message containing said version information (“After the computer receives the response including the URL of the upgrade package, the computer will send a query to the package server...” in col. 7 lines 4-6)

**Regarding claim 28:**

The rejection of claim 21 is incorporated, and further, Apfel et al disclose a management apparatus which sets and manages the one or more program components (“Auto-autoupdate is a cooperative scheduled client ‘pull’ routine designed to automatically update the...program module...” in col. 3 lines 39-41. The updating client is a management apparatus for the updating of the components.)

**Regarding claim 29 and 31:**

Claims 29 and 31 recite a product for performing the steps of claims 29 and 31 respectively, and are rejected for the reasons set for in connection with claims 29 and 31, respectively. The entirety of the Apfel et al patent describes a computer process for upgrading software, which would constitute a product for the computer system.

Art Unit: 2124

7. Claims 11, 18, 19, 20 and 30 are rejected under 35 U.S.C. 102(e) as being anticipated by U.S. Patent 6,493,768 to Boutcher.

**Regarding claim 11:**

Boutcher teaches:

- a distributed processing system (“a distributed computer system...” in col. 2 lines 62-63)
- including at least one client apparatus and a plurality of server apparatuses, and realizing an N-tier client-server environment (Note Figure 1 and the corresponding section of the disclosure)
- a client stub processing unit which executes stub processing of a first message which contains version information indicating a program version (“a version map, which is preferably resident in the client stub...” in col. 8 lines 35-36, and further, “the version of the server is compared against the different versions...for the client.” in col. 11 lines 20-23)
- a server skeleton processing unit which executes skeleton processing of a first message containing version information indicating a program version (“determines whether version mapping is required, generally by comparing the version of the server with the version of the client...” in col. 10 lines 54-57. A server skeleton would exist in the system for performing the actions of “marshalling, unmarshalling, sending and receiving data across a network in a distributed computer system” which Boutcher states is “generally understood in the art.” in col. 11 lines 8-11)
- a distributed-object storing repository which stores one or more distributed objects (“a plurality of mass storage devices...” in col. 4 line 54)

Art Unit: 2124

- a pre-transfer information management table which holds information on said one or more distributed objects stored in said distributed-object storing repository (“the supported versions of the server are compared with the version of the client...” in col. 13 lines 15-16. There must be a table of version information in the server if a comparison is to be made.)
- a memory which is allocated to an activated process, and temporarily stores at least one distributed object transferred from said distributed-object storing repository (“memory devices such as RAMs...the various applications...may be transferred or downloaded to a computer system...typically by first establishing a connection between the computer system and a server-type computer...” in col. 5 lines 34-46)
- a post-transfer information management table which holds information on said at least one distributed object stored in said memory (“the supported versions of the server are compared with the version of the client...” in col. 13 lines 15-16. There must be a table of version information in the server if a comparison is to be made, and further, once the update occurs, the table must be updated to reflect the current versions.)
- a distributed object execution control unit which dynamically links one of said one or more distributed objects corresponding to said version information contained in said first message of which skeleton processing is executed by said server skeleton processing unit, to said memory, so as to enable execution of a least one function in said one of said one or more distributed objects (“responds to the mapping by requesting...the second computer process to execute the second version of the one remote procedure.” in col. 15 lines 2-4. For the computer to execute the procedure, it must have been linked to a memory space.)
- a server stub processing unit which executes stub processing of a second message containing said version information so as to transmit the second message to another of said plurality of

Art Unit: 2124

server apparatuses (Note Figure 2B, item 44 and the corresponding section of the disclosure.)

**Regarding claim 18:**

The rejection of claim 11 is incorporated, and further, Boutcher discloses a management server which sets and manages one or more distributed objects stored in a distributed-object storing repository and the information on one or more distributed objects held in a pre-transfer information management table ("a server-type computer...which downloads or transfers a program product to other computers systems..." in col. 5 lines 49-51)

**Regarding claim 19:**

Boutcher teaches:

- a client apparatus ("a client computer..." in col. 2 line 49)
- an interface unit which controls operations of inputting and outputting software ("an operating system operating on CPU..." in col. 5 lines 5-6)
- a client stub processing unit which executes stub processing of a message which contains version information indicating a program version ("a version map, which is preferably resident in the client stub..." in col. 8 lines 35-36, and further, "the version of the server is compared against the different versions...for the client." in col. 11 lines 20-23)

**Regarding claim 20:**

Boutcher teaches:

- a server apparatus ("server systems..." in col. 4 line 59)

- a server skeleton processing unit which executes skeleton processing of a first message containing version information indicating a program version (“determines whether version mapping is required, generally by comparing the version of the server with the version of the client...” in col. 10 lines 54-57. A server skeleton would exist in the system for performing the actions of “marshalling, unmarshalling, sending and receiving data across a netowkr in a distributed computer system” which Boutilier states is “generally understood in the art.” in col. 11 lines 8-11)
- a distributed-object storing repository which stores one or more distributed objects (“a plurality of mass storage devices...” in col. 4 line 54)
- a pre-transfer information management table which holds information on said one or more distributed objects stored in said distributed-object storing repository (“the supported versions of the server are compared with the version of the client...” in col. 13 lines 15-16. There must be a table of version information in the server if a comparison is to be made.)
- a memory which is allocated to an activated process, and temporarily stores at least one distributed object transferred from said distributed-object storing repository (“memory devices such as RAMs...the various applications...may be transferred or downloaded to a computer system...typically by first establishing a connection between the computer system and a server-type computer...” in col. 5 lines 34-46)
- a post-transfer information management table which holds information on said at least one distributed object stored in said memory (“the supported versions of the server are compared with the version of the client...” in col. 13 lines 15-16. There must be a table of version information in the server if a comparison is to be made, and further, once the update occurs, the table must be updated to reflect the current versions.)

Art Unit: 2124

- a distributed object execution control unit which dynamically links one of said one or more distributed objects corresponding to said version information contained in said first message of which skeleton processing is executed by said server skeleton processing unit, to said memory, so as to enable execution of a least one function in said one of said one or more distributed objects ("responds to the mapping by requesting...the second computer process to execute the second version of the one remote procedure." in col. 15 lines 2-4. For the computer to execute the procedure, it must have been linked to a memory space.)
- a server stub processing unit which executes stub processing of a second message containing said version information so as to transmit the second message to another of said plurality of server apparatuses (Note Figure 2B, item 44 and the corresponding section of the disclosure.)

**Regarding claim 30:**

Claim 30 recites a product for performing the steps of claim 20, and is rejected for the reasons set forth in connection with claim 20. The entirety of the Boutcher patent describes a computer process for upgrading software, which would constitute a product for the computer system.

***Claim Rejections - 35 USC § 103***

8. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

Art Unit: 2124

9. Claims 2, 3, 6, 7, 22, 23, 26 and 27 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent 5,974,454 to Apfel et al in view of U.S. Patent 6,381,735 to Hunt.

**Regarding claim 2:**

The rejection of claim 1 is incorporated, and further, Apfel et al do not disclose a reference count which indicates the number of executions in which said each of said at least one program components are currently referring to. Hunt discloses in an analogous memory management system a reference count that indicates the number of executions in which said each of said at least one program components are currently referring to (“a component knows exactly how many clients have references to it...its reference count...” in col. 13 lines 34-35) It would have been obvious to someone of ordinary skill in the art at the time the invention was made to use the reference counter of Hunt with the component upgrading system of Apfel et al, as this would be used to ensure that the program component in the system disclosed by Apfel et al is not updated while the component is still in use.

**Regarding claim 3:**

The rejection of claim 2 is incorporated, and further, Apfel et al do not disclose removing one of said at least one program component from said program memory unit when a reference count for the program component is zero. Hunt discloses in an analogous memory management system removing a program component from a program memory unit when a reference count for the program component is zero (“When its reference count goes to zero, the component is responsible for freeing itself from memory.” in col. 13 lines 35-37) It would have been obvious to someone of ordinary skill in the art at the time the invention was made to use the reference counting system of

Hunt with the component upgrading system of Apfel et al, as this would recover memory space when it is no longer being used, thereby offering more memory to alternate components in the system disclosed by Apfel et al.

**Regarding claim 6:**

The rejection of claim 1 is incorporated, and further, note the rejection regarding claim 2. The evaluation count is identical to a reference count, as the reference counter is incremented when the system is evaluating the component the reference points to.

**Regarding claim 7:**

The rejection of claim 6 is incorporated, and further, note the rejection regarding claim 3.

**Regarding claim 22:**

The rejection of claim 21 is incorporated, and further, Apfel et al do not disclose a reference count which indicates the number of executions in which said each of said at least one program components are currently referring to. Hunt discloses in an analogous memory management system a reference count that indicates the number of executions in which said each of said at least one program components are currently referring to (“a component knows exactly how many clients have references to it...its reference count...” in col. 13 lines 34-35) It would have been obvious to someone of ordinary skill in the art at the time the invention was made to use the reference counter of Hunt with the component upgrading system of Apfel et al, as this would be used to ensure that the program component in the system disclosed by Apfel et al is not updated while the component is still in use.

**Regarding claim 23:**

The rejection of claim 22 is incorporated, and further, Apfel et al do not disclose removing one of said at least one program component from said program memory unit when a reference count for the program component is zero. Hunt discloses in an analogous memory management system removing a program component from a program memory unit when a reference count for the program component is zero ("When its reference count goes to zero, the component is responsible for freeing itself from memory." in col. 13 lines 35-37) It would have been obvious to someone of ordinary skill in the art at the time the invention was made to use the reference counting system of Hunt with the component upgrading system of Apfel et al, as this would recover memory space when it is no longer being used, thereby offering more memory to alternate components in the system disclosed by Apfel et al.

**Regarding claim 26:**

The rejection of claim 21 is incorporated, and further, note the rejection regarding claim 22. The evaluation count is identical to a reference count, as the reference counter is incremented when the system is evaluating the component the reference points to.

**Regarding claim 27:**

The rejection of claim 26 is incorporated, and further, note the rejection regarding claim 23.

10. Claims 4, 5, 24 and 25 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent 5,974,454 to Apfel et al in view of U.S. Patent 5,940,827 to Hapner et al.

**Regarding claim 4:**

The rejection of claim 1 is incorporated, and further, Apfel et al do not disclose an evaluation flag which indicates whether or not a program component is under evaluation. Hapner et al discloses in an analogous distributed computing system an evaluation flag which indicates whether or not a program component is under evaluation ("having a data structure which includes a true or false flag, thereby preventing all other threads from operating on this condition variable." in col. 10 lines 28-30). It would have been obvious to someone of ordinary skill in the art at the time the invention was made to use the thread locking system of Hapner et al with the component upgrading system of Apfel et al, as this would ensure that steps of the system occur in a desired order, as suggested by Hapner et al in col. 10 lines 21-22.

**Regarding claim 5:**

The rejection of claim 4 is incorporated, and further, Apfel et al do not disclose a program executing unit executing a program component for which said evaluation flag is contained in said message, in a separate process, when the evaluation flag indicates that said program component for which said evaluation flag is contained in said message is under evaluation. Hapner et al discloses an evaluation flag as claimed ("The first thread may then evaluate the true or false flag, performing the desired action if the flag value is true." in col. 10 lines 32-33.). It would have been obvious to someone of ordinary skill in the art at the time the invention was made to use the thread locking system of Hapner et al with the component upgrading system of Apfel et al, as this would ensure that steps of the system occur in a desired order, as suggested by Hapner et al in col. 10 lines 21-22.

**Regarding claim 24:**

The rejection of claim 21 is incorporated, and further, note the rejection regarding claim 4.

**Regarding claim 25:**

The rejection of claim 24 is incorporated, and further, note the rejection regarding claim 5.

11. Claims 12, 13, 16 and 17 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent 6,493,768 to Boutcher in view of U.S. Patent 6,381,735 to Hunt.

**Regarding claim 12:**

The rejection of claim 11 is incorporated, and further, Boutcher does not disclose a reference count which indicates the number of executions in which said each of said at least one program components are currently referring to. Hunt discloses in an analogous distributed system a reference count that indicates the number of executions in which said each of said at least one program components are currently referring to ("a component knows exactly how many clients have references to it...its reference count..." in col. 13 lines 34-35) It would have been obvious to someone of ordinary skill in the art at the time the invention was made to use the reference counter of Hunt with the version management system of Boutcher, as this would be used to ensure that the program component in the system disclosed by Boutcher is not updated or remapped while the component is still in use.

**Regarding claim 13:**

Art Unit: 2124

The rejection of claim 12 is incorporated, and further, Boutilcher does not disclose removing one of said at least one program component from said program memory unit when a reference count for the program component is zero. Hunt discloses in an analogous distributed system removing a program component from a program memory unit when a reference count for the program component is zero ("When its reference count goes to zero, the component is responsible for freeing itself from memory." in col. 13 lines 35-37) It would have been obvious to someone of ordinary skill in the art at the time the invention was made to use the reference counting system of Hunt with the version management system of Boutilcher, as this would recover memory space when it is no longer being used, thereby offering more memory to alternate processes in the system disclosed by Boutilcher.

**Regarding claim 16:**

The rejection of claim 11 is incorporated, and further, note the rejection regarding claim 12. The evaluation count is identical to a reference count, as the reference counter is incremented when the system is evaluating the component the reference points to.

**Regarding claim 17:**

The rejection of claim 16 is incorporated, and further, note the rejection of claim 13.

12. Claims 14 and 15 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent 6,493,768 to Boutilcher in view of U.S. Patent 5,940,827 to Hapner et al.

**Regarding claim 14:**

Art Unit: 2124

The rejection of claim 11 is incorporated, and further, Boutcher does not disclose an evaluation flag which indicates whether or not a program component is under evaluation. Hapner et al discloses in an analogous distributed computing system an evaluation flag which indicates whether or not a program component is under evaluation ("having a data structure which includes a true or false flag, thereby preventing all other threads from operating on this condition variable." in col. 10 lines 28-30). It would have been obvious to someone of ordinary skill in the art at the time the invention was made to use the thread locking system of Hapner et al with the version management system of Boutcher, as this would ensure that steps of the system occur in a desired order, as suggested by Hapner et al in col. 10 lines 21-22.

**Regarding claim 15:**

The rejection of claim 14 is incorporated, and further, Boutcher does not disclose a program executing unit executing a program component for which said evaluation flag is contained in said message, in a separate process, when the evaluation flag indicates that said program component for which said evaluation flag is contained in said message is under evaluation. Hapner et al discloses an evaluation flag as claimed ("The first thread may then evaluate the true or false flag, performing the desired action if the flag value is true." in col. 10 lines 32-33.). It would have been obvious to someone of ordinary skill in the art at the time the invention was made to use the thread locking system of Hapner et al with the version management system of Boutcher, as this would ensure that steps of the system occur in a desired order, as suggested by Hapner et al in col. 10 lines 21-22.

Art Unit: 2124

***Conclusion***

13. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.
14. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Trent J Roche whose telephone number is (703)305-4627. The examiner can normally be reached on Monday-Friday, 8:30 am - 5 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (703)305-9662. The fax phone number for the organization where this application or proceeding is assigned is (703) 872-9306.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703)305-3900.

Trent J Roche  
Examiner  
Art Unit 2124

TJR



**ANTONY NGUYEN-BA  
PRIMARY EXAMINER**